What is Claimed:

1. A virtualization system for a host computer having at least one host processor and system resources including memory divided into most privileged system memory and less privileged user memory, the system comprising:

virtualization software that operates in said less privileged user memory and divides said host computer into a plurality of virtual partitions including at least one user guest partition and at least one system partition, said at least one user guest partition providing a virtualization environment for at least one guest operating system, and said at least one system partition maintaining a resource database for use in managing use of said at least one host processor and said system resources;

at least one monitor that operates in said most privileged system memory and maintains guest applications in said at least one guest partition within memory space allocated by said at least one system partition to said at least one guest partition; and

a context switch between said at least one monitor and said respective guest and system partitions for controlling multitask processing of software in said partitions on said at least one host processor.

2. The virtualization system of claim 1, wherein said at least one system partition includes a resource management software application that assigns system resources to respective system and guest partitions and provides an index to the assigned system resource in said resource database.

3. The virtualization system of claim 2, wherein requested changes in the assignment of said system resources by said resource management software are communicated to said resource management software as transactions that are processed by said resource management software to update said resource database.

4. The virtualization system of claim 3, wherein said resource management software allocates shared memory to respective partitions as memory channels through which said transactions may pass from one partition to another.

5. The virtualization system of claim 4, wherein a system partition that experiences a processing failure is recovered by rebooting said failed system partition, reassigning system resources preserved for the failed system partition to the rebooted system partition and rolling

back any pending transactions in progress by said failed partition to reinstate a status of the resource database entries to a status prior to the time of failure of said system partition.

6. The virtualization system of claim 5, wherein said at least one system partition comprises an ultravisor partition that includes said resource database and said resource management software application and a command partition that includes resource allocation software that owns a resource allocation policy for said host system and creates transactions that pass through a command memory channel between said command partition and said ultravisor partition for processing by said resource management software for reallocation of said system resources as specified in said transaction.

7. The virtualization system of claim 6, wherein said resource management software maintains an audit log of processed transactions to enable said rolling back of transactions involving said failed system partition and enables reapplication of transactions from said audit log.

8. The virtualization system of claim 6, wherein said system partitions include a boot partition that contains hardware partition boot firmware for recovery operations when necessary to boot and reboot an other system partition and initiates transactions to said resource management software application requesting resources for said other system partition until the command partition may create transactions for allocating resources to the other system partition.

9. The virtualization system of claim 6, wherein said command partition include resource allocation software generates said transactions based on said resource allocation policy.

10. The virtualization system of claim 1, wherein said at least one monitor includes a lead monitor associated with a system partition containing said resource database and said lead monitor limits itself to read only access to partition descriptors stored in said resource database.

11. The virtualization system of claim 10, wherein said at least one monitor includes a monitor associated with a guest partition, and said lead monitor is configured to read partition descriptors for said guest partition from said resource database and to provide said partition descriptors to said monitor, whereby said monitor may use said partition descriptors to constrain guest applications within said guest partition.

12. The virtualization system of claim 11, wherein said monitor does not control access to any system resources.

13. The virtualization system of claim 11, wherein said at least one monitor includes a different monitor instance associated with each different guest partition, each said monitor being customized to a guest operating system in its corresponding guest partition so as to prevent said guest operating system from obtaining resources that have not been allocated to said guest partition by partition descriptors for said guest operating system in said resource database.

14. The virtualization system of claim 13, wherein when said guest operating system is programmed for an x86 processor, the associated monitor prevents execution by said guest operating system of sensitive operating system instructions that must be resolved using traps.

15. The virtualization system of claim 13, wherein some monitor instances have access to shared memory that is accessible by other monitor instances so as to allow the sharing of information among monitors associated with different partitions.

16. The virtualization system of claim 2, wherein said system resources include memory channels that enable the communication of transactions among respective guest and system partitions of said host computer, said memory channels comprising shared memory allocated to respective partitions.

17. The virtualization system of claim 16, wherein said at least one system partition includes at least one input/output (I/O) partition that maps physical I/O hardware of said host computer to endpoints of an I/O channel server in said I/O partition, said I/O channel server sharing the physical I/O hardware with at least one guest partition or another system partition via a memory channel between said I/O partition and said at least one guest partition or system partition, said resource management software allocating shared memory to said at least one guest partition or system partition and to said I/O partition to form said memory channel.

18. The virtualization system of claim 17, wherein upon receipt of a request to said I/O channel server from said at least one guest partition or system partition to access physical I/O hardware said I/O partition checks with partition descriptors stored in a monitor associated with said at least one guest partition or system partition to verify that the requested physical I/O hardware access is valid.

19. The virtualization system of claim 17, wherein said mapping by said at least one I/O partition of said physical I/O hardware of said host computer to endpoints of said I/O channel server in said I/O partition is performed by I/O partition software that multiplexes through shared common I/O physical hardware any I/O requests to said common I/O physical hardware from multiple partitions connected to said I/O partition by respective memory channels.

20. The virtualization system of claim 19, wherein an I/O monitor associated with said I/O partition implements a system call interface between said I/O monitor and said I/O partition, said system call interface converting and validating client partition relative addresses, obtained as buffer parameters of requests sent through respective memory channels from client memory channel drivers, as valid hardware physical addresses of memory currently assigned to the client partition requesting access to said common I/O physical hardware.

21. The virtualization system of claim 20, wherein messages between a server of said I/O partition and said respective client partitions are queued by the client partitions and de-queued by the I/O partition server and the partition relative physical addresses are converted by the I/O partition server to physical I/O hardware addresses with the aid of the I/O monitor, whereby data may be exchanged with hardware I/O adapters connected between said I/O monitor and said common I/O physical hardware.

22. The virtualization system of claim 17, wherein said mapping by said at least one I/O partition of said physical I/O hardware of said host computer to endpoints of said I/O channel server in said I/O partition is performed by passing I/O setup information via said memory channel to said I/O channel server so as to set up a high performance memory channel between a client partition requesting I/O access and intelligent physical I/O hardware and sending data directly between said client partition requesting I/O access and said intelligent physical I/O hardware via said high performance memory channel.

23. The virtualization system of claim 22, wherein the client partition requesting I/O access transfers data via said I/O memory channel to said intelligent physical I/O hardware using one of a user mode I/O or direct memory access data transfer operation.

24. The virtualization system of claim 17, wherein the at least one I/O partition includes two redundant I/O partitions.

25. The virtualization system of claim 2, wherein said system resources include input/output (I/O) hardware resources of said host computer.

26. The virtualization system of claim 2, wherein said system resources include whole or fractional parts of the processing of each host processor of said host computer.

27. The virtualization system of claim 26, wherein the resource management software application allocates processing cycles of said at least one host processor by limiting privilege levels of virtual processors so as to allow control of said at least one host processor through control of an Interrupt Descriptor Table (IDT) by said resource management software application.

28. The virtualization system of claim 27, wherein the resource management software application accesses the IDT to provide a timer interrupt to the at least one host processor, wherein said timer interrupt is used by said context switch to initiate virtual processor context switches.

29. The virtualization system of claim 2, wherein said system resources include physical memory of said host computer.

30. The virtualization system of claim 29, wherein a memory allocation page map of said resource database is organized according to a tiered page size model including a hierarchy of scales using 2^x as a scaling factor whereby an index page at each tiered page size level may allocate 2^x memory blocks at a size of the next lower tiered page size level.

31. The virtualization system of claim 30, wherein said resource management software application allocates memory to said respective system and guest partitions by storing a partition descriptor for a desired partition number [G,M,K] in said memory as $Mem(G,M,K) = ((G*2^{10}+M)*2^{10}+K)*2^{10}*$(word size), where word size is a power of 2.

32. The virtualization system of claim 31, wherein the stored partition descriptor is provided to the monitor associated with the partition defined by the partition descriptor whereby said monitor may constrain applications in its partition to the memory defined by said partition descriptor.

33. The virtualization system of claim 30, wherein x=10 and wherein a virtual partition number is represented in said memory allocation page map as a 32 bit index (2, 10, 10, 10) into a map of 4k pages that identifies the virtual partition descriptor for the virtual partition with said virtual partition number, where a first bit indicates suballocation in smaller pages and three successive 2^{10} values identify scaled pages.

34. The virtualization system of claim 2, wherein said at least one system partition comprises an ultravisor partition that includes said resource database and said resource management software application, a command partition that owns a resource allocation policy for said host system and creates transactions that pass through a command memory channel between said command partition and said ultravisor partition for processing by said resource management software for reallocation of said system resources as specified in said transaction.

35. The virtualization system of claim 34, wherein said at least one system partition further comprises an operations partition that owns a configuration policy and tracks persistence for respective domains to which each partition of said at least one host computer is assigned.

36. The virtualization system of claim 35, wherein the operations partition exchanges resource transactions with said command partition via a secure connection.

37. The virtualization system of claim 35, wherein the operations partition includes application software that maintains a persistent database of virtual partition definitions for at least one domain of said at least one host computer.

38. The virtualization system of claim 37, wherein the command partition stores a copy of the virtual partition definitions for said at least one domain for bootstrap purposes for initial startup and in the event of a partition failure or a hardware failure of a host computer.

39. The virtualization system of claim 37, wherein upon activation of a partition, the operations partition selects a host computer of said at least one host computer having required resources for said activated partition, connects to a resource service running in a command partition of said host computer, and provides a definition of the activated partition and a start command to the resource service.

40. The virtualization system of claim 39, wherein said command partition stores a copy of said resource database, uses said copy of said resource database to select appropriate resources for the activated partition, and creates a transaction to update said resource database via said command memory channel.

41. The virtualization system of claim 35, wherein said operations partition includes operations service software that monitors operation of said at least one host computer and, upon detection of host computer failure, chooses a new host computer for virtual partitions assigned to a failed host computer.

42. The virtualization system of claim 37, wherein said operations partition assigns an interconnected set of system resources of said at least one host computer to a zone and respective partitions are assigned to the zones with the system resources required by the respective partitions, where a zone is unit of resource allocation for system resources of said at least one host computer within a computer network.

43. The virtualization system of claim 42, wherein said operations partition assigns new partitions to a host computer that does not include said operations partition by sending, over a network connection, a resource transaction to a command partition of the host computer that does not include said operations partition.

44. The virtualization system of claim 43, wherein said operations partition enables migration of an active partition on a first host computer to a second host computer by transferring memory contents of the active partition from the first host computer to a target partition activated in the second host computer via said network connection.

45. The virtualization system of claim 35, wherein said configuration policy targets allocation of system resources to a zone based on at least one of quality of service guarantees, bandwidth guarantees, and physical location of respective host computers.

46. The virtualization system of claim 35, wherein said configuration policy is changeable by a user to permit changes in configuration of said system resources based on different system resource schedules at different times.

47. The virtualization system of claim 35, wherein said at least one system partition further comprises a redundant operations partition in a second host computer different from the host computer hosting said operations partition.

48. A method of managing a plurality of operating system instances on a host computer having at least one host processor and system resources, the method comprising the steps of:

dividing said host computer into a plurality of virtual partitions including at least one user guest partition and at least one system partition, said at least one user guest partition providing a virtualization environment for at least one guest operating system, and said at least one system partition maintaining a resource database for use in managing use of said at least one host processor and said system resources;

maintaining guest applications in said at least one guest partition within memory space allocated by said at least one system partition to said at least one guest partition; and

providing a context switch between said respective guest and system partitions for controlling multitask processing of software in said partitions on said at least one host processor.

49. The method of claim 48, wherein said at least one system partition includes a resource management software application that performs the steps of assigning system resources to respective system and guest partitions and providing an index to the assigned system resource in said resource database.

50. The method of claim 49, further comprising the step of communicating requested changes in the assignment of said system resources by said resource management software as transactions that may be processed by said resource management software to update said resource database.

51. The method of claim 50, further comprising the step of allocating shared memory to respective partitions as memory channels through which said transactions may pass from one partition to another.

52. The method of claim 51, further comprising the step of recovering a failed system partition by rebooting said failed system partition, reassigning system resources preserved for the failed system partition to the rebooted system partition, and rolling back any pending transactions in progress by said failed partition to reinstate a status of the resource database entries to a status prior to the time of failure of said system partition.

53. The method of claim 52, wherein said at least one system partition comprises an ultravisor partition that includes said resource database and said resource management software application and a command partition that includes resource allocation software that owns a resource allocation policy for said host system and that creates transactions for updating said resource database, comprising the further step of providing a command memory channel between said command partition and said ultravisor partition and passing said transaction through said command memory channel to said resource management software application for reallocation of said system resources as specified in said transaction.

54. The method of claim 53, comprising the further steps of providing an audit log of transactions processed by said resource management software application and, upon failure of a system partition, using said audit log to perform at least one of the steps of rolling back transactions involving said failed system partition to reinstate system resources at a time before failure of the system partition and reapplying completed transaction stored in the audit log.

55. The method of claim 53, further comprising the steps of recovering a failed system partition by rebooting the failed system partition, initiating transactions to said resource management software application requesting the resources of the failed system partition prior to failure, and reallocating the requested resources to the rebooted system partition.

56. The method of claim 53, comprising the step of separating resource management functionality into management by applications in at least three separate partitions, an operations partition that maintains resource allocation policy, a command partition that generates transactions requesting resources in accordance with the resource allocation policy, and a resource management partition that processes the transactions and updates the resource database based on said transaction processing.

57. The method of claim 48, wherein the step of maintaining guest applications within said allocated memory space comprises the step of providing a lead monitor associated with a system partition containing said resource database and limiting said lead monitor to read only access to partition descriptors stored in said resource database.

58. The method of claim 57, further comprising the steps of said lead monitor reading partition descriptors for said guest partition from said resource database, providing said partition

descriptors to a monitor associated with said guest partition, and said monitor using said partition descriptors to constrain guest applications within said guest partition.

59. The method of claim 58, further comprising the steps of providing a different monitor instance associated with each different guest partition and customizing each said monitor to a guest operating system in its corresponding guest partition so as to prevent said guest operating system from obtaining resources that have not been allocated to said guest partition by partition descriptors for said guest operating system in said resource database.

60. The method of claim 59, wherein when said guest operating system is programmed for an x86 processor, the associated monitor performing the step of preventing execution by said guest operating system of sensitive operating system instructions that must be resolved using traps.

61. The method of claim 59, further comprising the step of providing some monitor instances with access to shared memory that is accessible by other monitor instances so as to allow the sharing of information among monitors associated with different partitions.

62. The method of claim 49, wherein said system resources include memory channels that enable the communication of transactions among respective guest and system partitions of said host computer, and said step of assigning system resources includes the step of assigning said memory channels to respective partitions.

63. The method of claim 62, comprising the further steps of mapping physical I/O hardware of said host computer to endpoints of an I/O channel server in an input/output (I/O) partition, said I/O channel server sharing the physical I/O hardware with at least one guest partition or another system partition via a memory channel between said I/O partition and said at least one guest partition or system partition, said resource management software allocating shared memory to said at least one guest partition or system partition and to said I/O partition to form said memory channel.

64. The method of claim 63, wherein, upon receipt of a request to said I/O channel server from said at least one guest partition or system partition to access physical I/O hardware, said I/O partition performs the step of checking with partition descriptors associated with said at least one guest partition or system partition to verify that the requested physical I/O hardware access is valid.

65. The method of claim 63, wherein said mapping by said at least one I/O partition of said physical I/O hardware of said host computer to endpoints of said I/O channel server in said I/O partition comprises the step of multiplexing through shared common I/O physical hardware any I/O requests to said common I/O physical hardware from multiple partitions connected to said I/O partition by respective memory channels.

66. The method of claim 65, further comprising the steps of implementing a system call interface between an I/O monitor associated with said I/O partition and said I/O partition and said system call interface converting and validating client partition relative addresses, obtained as buffer parameters of requests sent through respective memory channels from client memory channel drivers, as valid hardware physical addresses of memory currently assigned to the client partition requesting access to said common I/O physical hardware.

67. The method of claim 66, comprising the further steps of respective client partitions queuing messages between a server of said I/O partition and said respective client partitions, the I/O partition server de-queuing the partition relative physical addresses and converting the partition relative physical addressed to physical I/O hardware addresses, and exchanging data with hardware I/O adapters connected between said I/O monitor and said common I/O physical hardware.

68. The method of claim 63, wherein said mapping by said at least one I/O partition of said physical I/O hardware of said host computer to endpoints of said I/O channel server in said I/O partition comprises the steps of passing I/O setup information via said memory channel to said I/O channel server so as to set up a high performance memory channel between a client partition requesting I/O access and intelligent physical I/O hardware and sending data directly between said client partition requesting I/O access and said intelligent physical I/O hardware via said high performance memory channel.

69. The method of claim 68, wherein the step of sending data directly between said client partition requesting I/O access and said intelligent physical I/O hardware comprises the step of performing one of a user mode I/O and a direct memory access data transfer operation.

70. The method of claim 49, wherein the step of assigning system resources comprises the step of allocating processing cycles of said at least one host processor by limiting privilege levels

of virtual processors so as to allow control of said at least one host processor through control of an Interrupt Descriptor Table (IDT) by said resource management software application.

71. The method of claim 70, further comprising the steps of providing a timer interrupt to the at least one host processor and using said timer interrupt to initiate virtual processor context switches.

72. The method of claim 49, further comprising the steps of organizing a memory allocation page map of said resource database according to a tiered page size model including a hierarchy of scales using 2^x as a scaling factor and an index page at each tiered page size level allocating 2^x memory blocks at a size of the next lower tiered page size level.

73. The method of claim 72, further comprising the step of allocating memory to said respective system and guest partitions by storing a partition descriptor for a desired partition number [G,M,K] in said memory as $Mem(G,M,K) = ((G*2^{10} +M)*2^{10}+K)*2^{10}*(\text{word size})$, where word size is a power of 2.

74. The method of claim 73, further comprising the steps of providing the stored partition descriptor to a monitor associated with the partition defined by the partition descriptor and said monitor constraining applications in its partition to the memory defined by said partition descriptor.

75. The method of claim 72, wherein x=10, further comprising the step of representing a virtual partition number in said memory allocation page map as a 32 bit index (2, 10, 10, 10) into a map of 4k pages that identifies the virtual partition descriptor for the virtual partition with said virtual partition number, where a first bit indicates suballocation in smaller pages and three successive 2^{10} values identify scaled pages.

76. The method of claim 49, wherein said at least one system partition comprises an ultravisor partition that includes said resource database and said resource management software application and a command partition that owns a resource allocation policy for said host system, further comprising the steps of creating transactions that pass through a command memory channel between said command partition and said ultravisor partition and said resource management software processing said transaction for reallocation of said system resources as specified in said transaction.

77. The method of claim 76, wherein the steps of assigning system resources comprises the step of assigning each partition of said at least one host computer to a domain based on a configuration policy.

78. The method of claim 77, further comprising the step of maintaining a persistent database of virtual partition definitions for at least one domain of said at least one host computer.

79. The method of claim 78, further comprising the step of storing a copy of virtual partition definitions for said at least one domain for bootstrap purposes for initial startup and in the event of a partition failure or a hardware failure of a host computer.

80. The method of claim 78, wherein upon activation of a partition, performing the steps of selecting a host computer of said at least one host computer having required resources for said activated partition, connecting to a resource service running in a command partition of said host computer, and providing a definition of the activated partition and a start command to the resource service.

81. The method of claim 80, further comprising the step of storing a copy of said resource database in said command partition, using said copy of said resource database to select appropriate resources for the activated partition, and creating a transaction to update said resource database via said command memory channel.

82. The method of claim 76, further comprising the steps of monitoring operation of said at least one host computer and, upon detection of host computer failure, choosing a new host computer for virtual partitions assigned to a failed host computer.

83. The method of claim 78, further comprising the steps of assigning an interconnected set of system resources of said at least one host computer to a zone and assigning respective partitions to the zones with the system resources required by the respective partitions, where a zone is unit of resource allocation for system resources of said at least one host computer within a computer network.

84. The method of claim 83, wherein said partitions assigning step comprises the steps of assigning new partitions to a host computer by sending, over a network connection, a resource transaction to a command partition of the host computer that is to host the new partition.

85. The method of claim 84, further comprising the step of migrating an active partition on a first host computer to a second host computer by transferring memory contents of the active partition from the first host computer to a target partition activated in the second host computer via said network connection.

86. The method of claim 77, further comprising the step of changing said configuration policy based on different system resource schedules at different times.